Amendments to the Claims:

1. (Currently amended) A computer-implemented method for dynamically modifying an executing heterogeneous program in a distributed computing environment, the method comprising:

obtaining a system reference to a an online target system on which the heterogeneous program is executing;

obtaining a program reference to the heterogeneous program based on the system reference;

locating a component of the heterogeneous program based on the program reference, the component residing in a target system memory associated with the target system;

creating a modified executable code based on an internal representation of the component derived from an original executable code associated with the component; and

inserting the modified executable code into the online target system memory without taking the online target system offline.

2. (Original) The computer-implemented method of Claim 1, wherein the modified executable code comprises a user mode code that executes in user mode.

3. (Original) The computer-implemented method of Claim 2, wherein inserting the modified executable code comprises:

suspending one or more threads from processing on the target system;

if the modified executable code consumes more memory than the original executable code, injecting the modified executable code into the target system memory at a new memory location;

else, patching the modified executable code into the target system memory by overwriting an original memory area with the modified executable code, the original executable code being resident in the original memory area; and

resuming the one or more threads for processing on the target system.

4. (Original) The computer-implemented method of Claim 3, further comprising fixing a first thread out of the one or more threads if the first thread was suspended while executing a portion of the original executable code in the original memory area.

5. (Original) The computer-implemented method of Claim 3, wherein injecting the modified executable code comprises:

creating a copy of the original executable code;

locating the new memory location for the modified executable code;

writing the modified executable code to the target memory at the new memory location; and

redirecting execution of the heterogeneous component to the modified executable code.

6. (Original) The computer-implemented method of Claim 5, wherein redirecting execution includes writing a jump instruction in a first address of the original memory area, the jump instruction including an offset to the new memory location.

7. (Original) The computer-implemented method of Claim 1, wherein the modified executable code comprises a kernel mode code that executes in kernel mode.

8. (Original) The computer-implemented method of Claim 7, wherein inserting the modified executable code comprises:

replacing a first portion of the original executable code that resides in a first part of the original memory area with an instruction that disallows a thread from executing instructions in a second part of the original memory area;

replacing the second part of the original memory area with a portion of the modified executable code; and

replacing the instruction in the first part of the original memory area with another portion of the modified executable code, in manner such that the original memory area contains the modified executable code.

9. (Previously presented) The computer-implemented method of Claim 1, further comprising determining whether the target system is a remote system, and if the target system is a remote system, initiating a dynamic instrumentation process on the target system that enables communication with a tool residing on a local system that is performing the dynamic modifications to the heterogeneous program.

10. (Original) The computer-implemented method of Claim 1, wherein the internal representation is derived from the original executable code that resides in the target system memory.

11. (Original) The computer-implemented method of Claim 1, wherein the internal representation is derived from the original executable code that resides on a local storage device.

12. (Original) The computer-implemented method of Claim 1, wherein the modified executable code comprises a procedure.

13. (Original) The computer-implemented method of Claim 1, wherein the modified executable code comprises a basic block.

14. (Original) The computer-implemented method of Claim 1, wherein the modified executable code comprises an instruction.

15. (Currently amended) A computerized system for modifying a heterogeneous program associated with an online target system without taking the target system offline, the system comprising:

a processing unit;

a system memory coupled to the processing unit through a system bus;

a computer-readable medium coupled to the processing unit through a system bus;

a hierarchical intermediate representation for a heterogeneous program residing in the system memory;

a transformation process executing in the processing unit for modifying the hierarchical intermediate representation to create a modified intermediate representation associated with the heterogeneous program;

a dynamic modification process executing in the processing unit for modifying an executable code in a target system memory based on the modified intermediate representation without taking the target system offline, the executable code being associated with the heterogeneous program.

16. (Original) The computerized system of Claim 15, wherein modifying the executable code in the target system includes:

suspending one or more threads from processing on the target system;

if a modified executable code based on the modified intermediate representation consumes more memory than the executable code, injecting the modified executable code into the target system memory at a new memory location;

else, patching the modified executable code into the target system memory by overwriting an original memory area with the modified executable code, the original memory are being associated with the executable code; and

resuming the one or more threads for processing on the target system.

17. (Original) The computerized system of Claim 16, wherein injecting the modified executable code includes:

creating a copy of the executable code;

locating the new memory location for the modified executable code;

writing the modified executable code to the target memory at the new memory location; and

redirecting execution of the heterogeneous program to the modified executable code.

18. (Original) The computerized system of Claim 15, wherein modifying the executable code in the target system includes:

replacing a first portion of the executable code that resides in a first part of the original memory area with an instruction that disallows a thread from executing instructions in a second part of the original memory area;

replacing the second part of the original memory area with a portion of the modified executable code; and

replacing the instruction in the first part of the original memory area with another portion of the modified executable code, in manner such that the original memory area contains the modified executable code.

19. (Original) The computer system of Claim 15, wherein the target system is a remote system.

20. (Currently amended) A computer-readable medium having computer-executable components for modifying an online target system without taking the target system offline, comprising:

a transformation process configured to modify an hierarchical intermediate representation of a heterogeneous program executing in a target system memory; and

a dynamic modification process configured to modify an executable code in the target system memory based on the modified intermediate representation without taking the target system offline, the executable code being associated with the heterogeneous program.